WHAT IS CLAIMED IS:

1. A processor comprising:

an instruction fetching circuit for calculating a lower portion of an effective address for an instruction word with a displacement, for replacing a value of the displacement in the instruction word according to the calculating result, and for storing the value in a storage circuit; and

the storage circuit for temporarily storing the instruction word, from which the stored instruction word is read at an instruction execution time,

wherein the effective address of said instruction word in the storage circuit is specified relative to a current value of a program counter address at the instruction execution time with the displacement, and

wherein said processor utilizes the value stored in said storage as a lower portion of the effective address at the instruction execution time.

2. The processor of claim 1,

wherein said storage circuit has additional storage areas each of which one-on-one corresponds to each said instruction word, and

wherein said storage circuit stores the calculating result in said additional storage areas.

3. The processor of claim 1, wherein the effective address is a branch target address.

4. The processor of claim 2, wherein the effective address is a branch target address.

5. The processor of claim 1, wherein said storage circuit is a cache or a buffer.

6. A processor comprising:

a storage circuit for temporarily storing an instruction word, from which the stored instruction word is read at an instruction execution time;

a decoder for receiving the instruction word and for determining whether an effective address of said instruction word is specified as a PC relative displacement value;

an adder for adding of the PC relative displacement value and predetermined lower bits of the PC address, and for outputting the calculating result as a portion of the effective address if said instruction word has the PC relative displacement value; and

a selector for replacing the displacement value in the instruction word with the calculating result outputted from said adder, and for outputting said replaced result to said storage circuit as a semiABS displacement value of the instruction, if said instruction word has the PC relative displacement value.

7. The processor of claim 6, further comprising an effective address calculator for receiving a portion of the instruction word that has been stored in said storage circuit, for performing a sign bit extension thereof, and for calculating the effective address by using the semiABS displacement value at the instruction execution time.

8. The processor of claim 6, wherein said storage circuit includes an area for storing a carry bit from said adder corresponding to the instruction word.

9. The processor of claim 8, further comprising:

an effective address calculator for receiving a portion of the instruction word that has been stored in said storage circuit, for performing a sign bit extension thereof, and for calculating the effective address with the semiABS displacement value at the instruction execution time,

wherein said effective address calculator calculates said effective address with said sign bit extension by using the carry bit stored in said storage circuit.

10. The processor of claim 6, wherein said storage circuit is an instruction cache.

11. The processor of claim 6, further comprising:

an effective address calculator for calculating the effective address with the semiABS displacement value at the instruction execution time; and

a carry bit recovering means for recovering the disregarded carry bit of said adder, including:

a comparator for comparing the semiABS displacement value and lower bits of the PC address;

a decoder for receiving the comparing result, a sign_bit of the semiABS displacement value, and a bit which is a digit higher than the highest bit of the lower bits of the PC address in the PC address thereby outputting a selecting signal for selecting one of +1, 0, and –1 according to a predetermined conversion table; and

means for adding +1, 0, or −1 to the rest bits of the PC address then to a upper portion of the effective address with a bit number equal to the rest bits of the PC address according to the selecting signal thereby deciding a recovered carry bit,

wherein said effective address calculator calculates said effective address without using the carry bit generated by said adder.

12. A method for converting a first instruction word with a PC relative displacement value into a second instruction word with a semiABS displacement value, comprising:

calculating a the semiABS displacement value by adding predetermined lower bits of a PC address and the PC relative displacement value;

replacing the PC relative displacement value in the first instruction word with the calculating result; and

storing the second instruction word with the semiABS displacement value in a storage circuit,

whereby the semiABS displacement value stored in said storage circuit is then immediately used as a portion of an effective address at the instruction execution time.

13. The method according to claim 12, further comprising:

receiving a portion of the instruction word stored in said storage circuit;

performing a sign bit extension without using a carry bit carried from the adding result; and

calculating the effective address with the semiABS displacement value at the instruction execution time.

14. A method for recovering a disregarded carry bit generated during the method according to claim 13, including:

comparing the semiABS displacement value and lower bits of the PC address;

receiving the comparing result, a sign bit of the semiABS displacement value, and a bit which is a digit higher than the highest hit of the lower bits of the PC address in the PC address thereby outputting a selecting signal for selecting one of +1, 0, and –1 according to a predetermined conversion table; and

adding +1, 0, or –1 to the rest bits of the PC address then to a upper portion of the effective address with a bit number equal to the rest bits of the PC address according to the selecting signal thereby recovering a carry bit carried from the most significant bit of the add step.

15. A method for recovering a disregarded carry bit generated during the method according to claim 13, including:

comparing the semiABS displacement value and lower bits of the PC address to obtain a carry bit carried from the adding step;

performing an exclusive OR operation on the carry bit, a sign bit of the semiABS displacement value, and the bit which is a digit higher than the highest hit of the lower bits of the PC address in the PC address thereby outputting a sign bit of the PC relative displacement value of the first instruction word; and

adding the carry bit, the sign bit, and the bit which is a digit higher than the highest hit of the lower bits of the PC address in the PC address to recover a carry bit carried from the adding step.